

Il calcolatore

Prima parlare della programmazione, e' bene fare una brevissima introduzione su come sono strutturati i **calcolatori elettronici**.

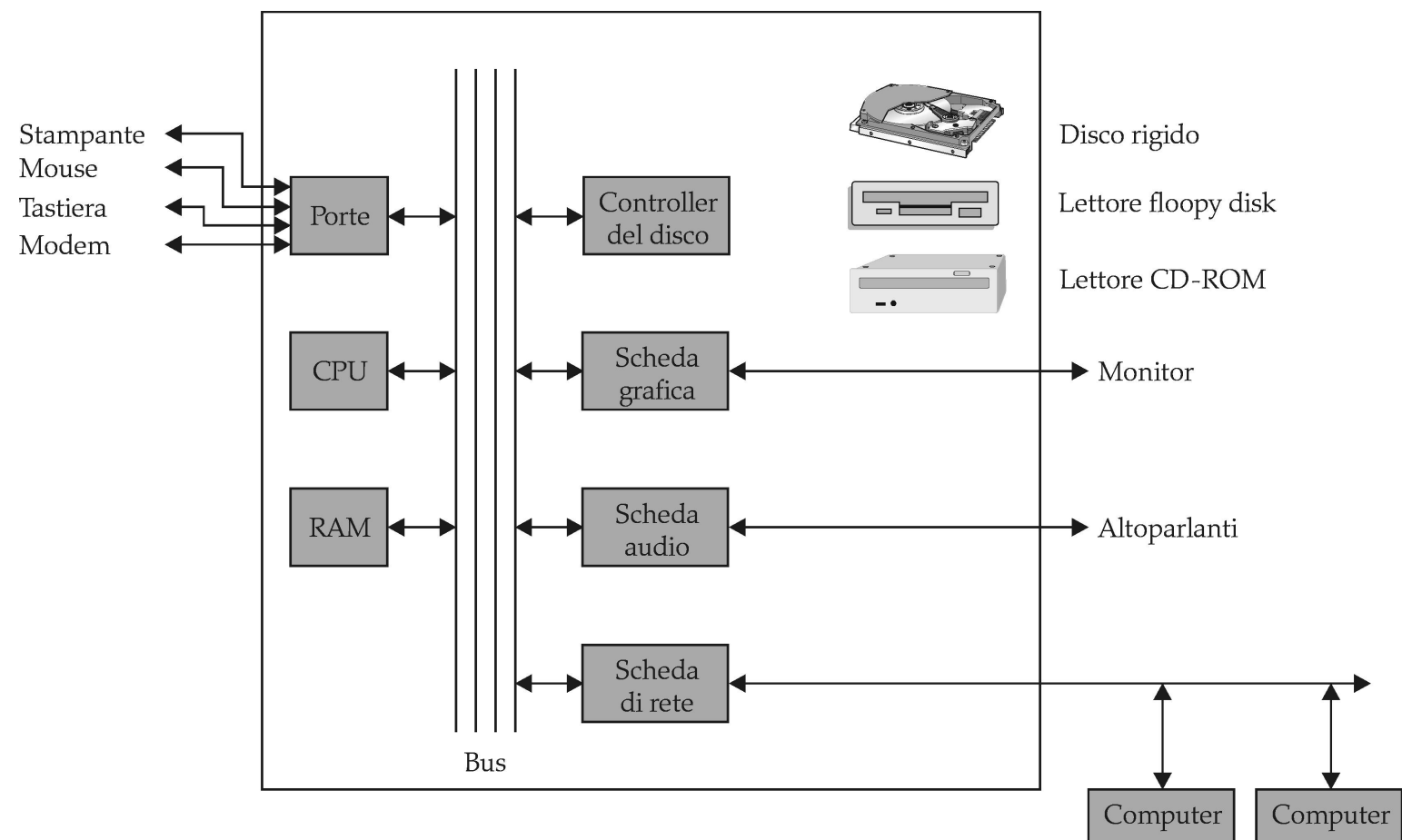
I calcolatori elettronici sono stati progettati e costruiti per la prima volta durante la Seconda Guerra Mondiale. Il primo calcolatore fu l'**ENIAC**, progettato e costruito all'Universita' della Pennsylvania nel periodo 1943 - 1946.

Nei calcolatori si distinguono due componenti fondamentali:

- l'**Hardware**, che e' costituito dalla parte elettronica (es. circuiti integrati) e meccanica (es. bracci che spostano le testine dell'Hard disk) del calcolatore;
- il **Software**, che e' costituito dall'insieme dei programmi che permettono di usare un calcolatore.

I calcolatori possono eseguire con grande velocita' (es. 2 Ghz = 2.000.000.000 di operazioni elementari al secondo) delle operazioni elementari. Vista questa enorme velocita', i calcolatori possono eseguire un numero impressionante di operazioni in pochi secondi. Questa capacita' di eseguire tantissime operazioni in pochi secondi e' alla base della enorme potenza di calcolo dei calcolatori moderni.

Architettura di un calcolatore (Hardware)



Rappresentazione schematica dell'hardware di un computer

I Programmi (Software)

I programmi servono per fare funzionare l'hardware (parte fisica) del computer.

Informalmente, un programma può essere definito come una serie di operazioni elementari che il calcolatore esegue in sequenza, e che trasforma un insieme di dati di ingresso (INPUT) in un insieme di dati di uscita (OUTPUT).

Esempio:

Un programma per fare la somma di due numeri prende i due numeri come dati di ingresso (es. 10 e 5), e ne restituisce la somma (es. 15) come dato di uscita

L'insieme dei programmi (es. Word Processor, Browser per Internet, etc.) e dei dati (es. i documenti scritti con il Word Processor, fotografie, etc.) presenti sul calcolatore ne costituiscono il Software (parte logica).

Un calcolatore può eseguire direttamente (sull'hardware) solo i programmi che sono scritti in linguaggio macchina, che è costituito da una sequenza di numeri binari.

In un calcolatore si distinguono vari "strati" di software.

Per semplicità ne consideriamo due:

- Il sistema operativo (come Linux, Windows, Mac Os, ...)
- Le applicazioni (come editor di testo, sistemi di gestione di basi di dati, giochi,...)

Cosa è la programmazione?

La programmazione è l'attività di sviluppare programmi per un calcolatore.

Lo scopo della scrittura di un programma è la risoluzione di un problema.

Per risolvere un problema bisogna:

- Formulare il problema (*specificazione dei requisiti*) in modo più o meno formale
- Capire il problema e scomporlo in parti gestibili (*analisi del problema*)
- Progettare una soluzione (*algoritmo*)
- Implementare la soluzione (*scrittura del codice*)
- Testare la soluzione e correggere eventuali errori (*verifica del programma, testing e debugging*)
- Tenere sempre aggiornato il programma (*manutenzione*)

Di tutte queste fasi vedremo nel corso principalmente quella di progettazione ed implementazione di una soluzione, e marginalmente quelle precedenti.

Gli Algoritmi

Un **algoritmo** per la risoluzione di un problema è una sequenza di passi

- discreti
- di lunghezza finita
- deterministici (dopo ogni passo si sa precisamente qual è il prossimo)
- ripetibili

che producono la soluzione del problema.

Esempio di algoritmi:

- procedimenti per calcolare il risultato di espressioni aritmetiche (operazioni elementari come addizione, sottrazione,...; o più complesse come radice quadrata,...) o algebriche (prodotti di polinomi, ...)

I Linguaggi di Programmazione

Per scrivere il programma che realizza un algoritmo, il programmatore deve usare un **linguaggio di programmazione**.

Un linguaggio di programmazione è costituito da un insieme di **parole** e di **simboli** e un **insieme di regole per combinarli**.

Se il programma rispetta queste regole, può essere tradotto (**compilato**) in linguaggio macchina, e quindi eseguito sul calcolatore.

A differenza dei linguaggi naturali, le regole dei linguaggi di programmazione (che sono **linguaggi formali**) sono molto rigide, per garantire l'assenza di ambiguità e la rapidità della compilazione.

Ci sono centinaia di linguaggi di programmazione, divisi in varie categorie:



Linguaggi
ad alto
livello
(C, Java,
Basic,
etc.)

Linguaggi
assembly

Linguaggi
macchina

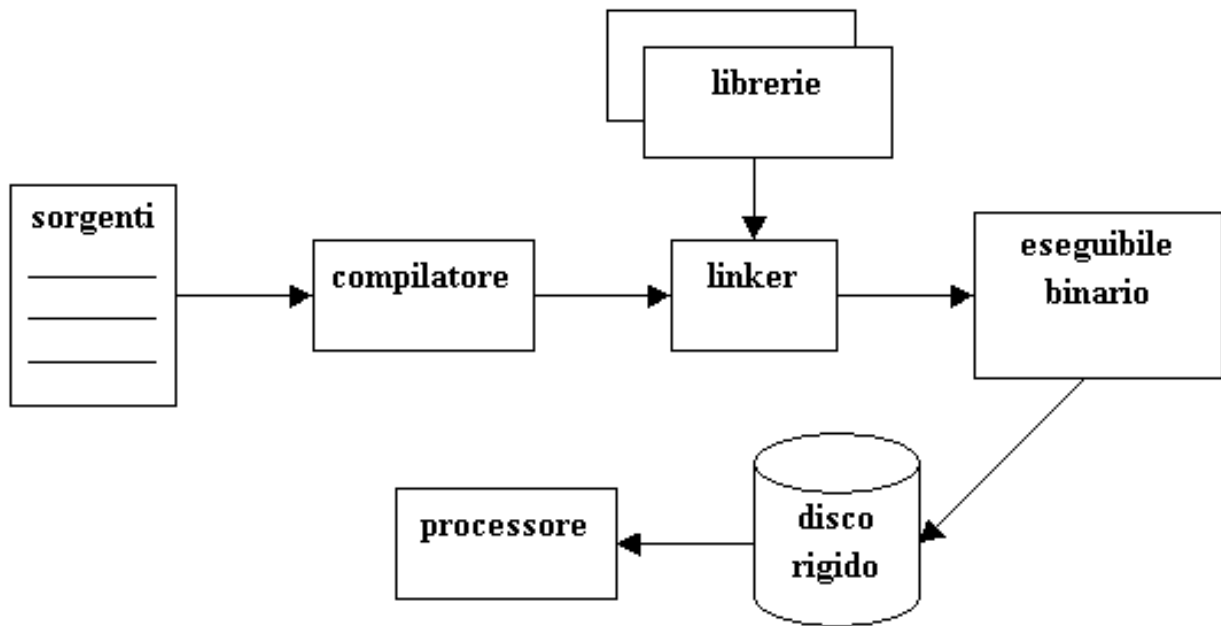


I passi necessari per creare ed eseguire un programma

I programmi scritti in un linguaggio ad alto livello non possono essere eseguiti direttamente dal calcolatore, che "capisce"

solo il linguaggio macchina.

Il programma (**codice sorgente**) viene tradotto in linguaggio macchina (**codice oggetto**) da un altro programma, il **compilatore**.



Dalla scrittura del programma alla sua esecuzione

Quindi l'attività di programmazione fa uso di **tool** (strumenti, o applicazioni) quali un **text editor** per la scrittura di programmi, e un compilatore.

Quali strumenti useremo per scrivere i nostri programmi in C?

Per scrivere un programma in C servono:

- un **editor di testo**
- un **compilatore**

Gli strumenti che useremo nel corso sono:

- **EMACS** (editor di testo)
- **GCC** (compilatore C)

L'ambiente operativo che useremo e' **LINUX**

Il Linguaggio C

Nel 1972 Dennis Ritchie progettava e realizzava la prima versione del linguaggio C presso i Bell Laboratories (su DEC PDP-11), ispirato da:

- linguaggio BCPL di Martin Richards (1967, usato per S.O. e compilatori),

- linguaggio B di Ken Thompson (autore del sistema operativo UNIX, 1970, su DEC PDP-7)

Nel 1983-1990 si arriva alla definizione dell'ANSI C (standard ISO)

BCPL e B erano linguaggi *non tipizzati* ed il C si distingueva nell'implementare una vasta gamma di tipi di dati (carattere, interi, numeri in virgola mobile, strutture).

Successivamente gli stessi Ritchie e Thompson riscrissero in C il codice di UNIX e oggi giorno sono scritti in C (o C++) quasi tutti i S.O. di ultima generazione.

Il C è indipendente dall'hardware e favorisce la *portabilità* del codice. Pur essendo un linguaggio ad alto livello permette operazioni di basso livello tipiche del linguaggio macchina.

E gli altri linguaggi? (Pascal, ADA, C++, Java, C#...)

[Torna alla HomePage del corso](#)